Table of Contents

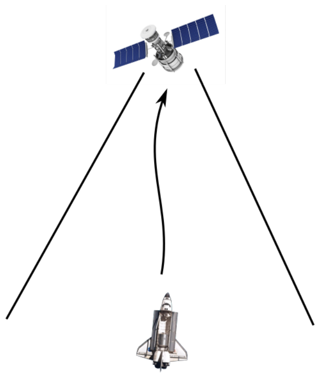
Verification of satellite rendezvous problem via SReachSet This example will

```
%demonstrate the use of SReachTools in verification of stochastic
%continuous-state discrete-time linear time-invariant (LTI) systems.
%
% Specifically, we will discuss how SReachTools can use Fourier
 transforms
% (<http://www.math.wsu.edu/faculty/genz/software/matlab/qsimvnv.m
 Genz's
% algorithm> and MATLAB's patternsearch), convex chance constraints,
 and
% Lagrangian methods to construct underapproximative stochastic reach
 sets.
%
% Our approaches is grid-free and recursion-free resulting in highly
 scalable
% solutions, especially for Gaussian-perturbed LTI systems.
%
% This Live Script is part of the SReachTools toolbox. License for the
 use of
% this function is given in
% <https://github.com/unm-hscl/SReachTools/blob/master/LICENSE
% https://github.com/unm-hscl/SReachTools/blob/master/LICENSE>.

% Prescript running
close all;
% clc;
clearvars;
srtinit
```

Problem formulation: Spacecraft motion via CWH dynamics

We consider both the spacecrafts, referred to as the deputy spacecraft and the chief spacecraft, to be in the same circular orbit. In this example, we will consider the problem of verification for the spacecraft rendezvous problem, i.e., identify all the initial states from which the deputy can can rendezvous with the chief while staying within the line-of-sight cone with a likelihood above a user-specified threshold.



Dynamics model for the deputy relative to the chief spacecraft The relative

```
%planar dynamics of the deputy with respect to the chief are described
 by the
%<https://doi.org/10.1109/CDC.2013.6760626 Clohessy-Wiltshire-Hill
 (CWH)
%equations,>
%
% $$\ddot{x} - 3 \omega x - 2 \omega \dot{y} = \frac{F_{x}}{m_{d}}$$
%
% $$            \ddot{y} + 2 \omega \dot{x} = \frac{F_{y}}{m_{d}}$$
%
% where the position of the deputy relative to the chief is $x,y \in
% \mathbf{R}$, $\omega = \sqrt{\frac{\mu}{R_{0}^{3}}}$ is the orbital
 frequency,
% $\mu$ is the gravitational constant, and $R_{0}$ is the orbital
 radius of the
% chief spacecraft.  We define the state as $\overline{x} = {[x\ y\
 \dot{x}\
% \dot{y}]}^\top \in \mathbf{R}^{4}$ which is the position and
 velocity of the
% deputy relative to the chief along $\mathrm{x}$- and $\mathrm{y}$-
 axes, and
% the input as $\overline{u} = {[F_{x}\ F_{y}]}^\top \in
% \mathcal{U}\subset\mathbf{R}^{2}$.
%
% We will discretize the CWH dynamics in time, via zero-order hold, to
 obtain
```

```
% the discrete-time linear time-invariant system and add a Gaussian
 disturbance
% to account for the modeling uncertainties and the disturbance
 forces,
%
% $$\overline{x}_{k+1} = A \overline{x}_{k} + B \overline{u}_{k} +
% \overline{w}_{k}$$
%
% with $\overline{w}_{k} \in \mathbf{R}^{4}$ as an IID Gaussian zero-
mean
% random process with a known covariance matrix $
\Sigma_{\overline{w}}$.
```

System definition

```
umax = 0.1;
mean_disturbance = zeros(4,1);
covariance_disturbance = diag([1e-4, 1e-4, 5e-8, 5e-8]);
% Define the CWH (planar) dynamics of the deputy spacecraft relative
 to the
% chief spacecraft as a LtiSystem object
sys = getCwhLtiSystem(4, Polyhedron('lb', -umax*ones(2,1), ...
                                        'ub',  umax*ones(2,1)), ...
       RandomVector('Gaussian',
 mean_disturbance,covariance_disturbance));
```

Methods to run

```
ft_run = 1;
cc_open_run = 1;
```

Target tube construction --- reach-avoid specification

```
time_horizon = 5;          % Stay within a line of sight cone for 4
 time steps and
                         % reach the target at t=5% Safe Set --- LoS
 cone
% Safe set definition --- LoS cone |x|<=y and y\in[0,ymax] and |vx|
<=vxmax and
% |vy|<=vymax
ymax = 2;
vxmax = 0.5;
vymax = 0.5;
A_safe_set = [1, 1, 0, 0;
             -1, 1, 0, 0;
              0, -1, 0, 0;
              0, 0, 1,0;
              0, 0,-1,0;
              0, 0, 0,1;
```

```
                0, 0, 0,-1];
b_safe_set = [0;
              0;
              ymax;
              vxmax;
              vxmax;
              vymax;
              vymax];
safe_set = Polyhedron(A_safe_set, b_safe_set);
% Target set --- Box [-0.1,0.1]x[-0.1,0]x[-0.01,0.01]x[-0.01,0.01]
target_set = Polyhedron('lb', [-0.1; -0.1; -0.01; -0.01], ...
                        'ub', [0.1; 0; 0.01; 0.01]);
target_tube = Tube('reach-avoid',safe_set, target_set, time_horizon);
slice_at_vx_vy = zeros(2,1);
```

Preparation for set computation

```
prob_thresh = 0.8;

n_dir_vecs = 10;
theta_vec = linspace(0, 2*pi, n_dir_vecs);
set_of_dir_vecs_ft = [cos(theta_vec);
                      sin(theta_vec);
                      zeros(2,n_dir_vecs)];
n_dir_vecs = 40;
theta_vec = linspace(0, 2*pi, n_dir_vecs);
set_of_dir_vecs_cc_open = [cos(theta_vec);
                           sin(theta_vec);
                           zeros(2,n_dir_vecs)];
init_safe_set_affine = Polyhedron('He',[zeros(2,2) eye(2,2)
 slice_at_vx_vy]);
```

CC (Linear program approach)

```
if cc_open_run
    options = SReachSetOptions('term', 'chance-open', ...
        'set_of_dir_vecs', set_of_dir_vecs_cc_open, ...
        'init_safe_set_affine', init_safe_set_affine, ...
        'verbose', 1);
    timer_cc_open = tic;
    [polytope_cc_open, extra_info] = SReachSet('term','chance-open',
 sys, ...
        prob_thresh, target_tube, options);
    elapsed_time_cc_open = toc(timer_cc_open);
end
```

```
Computing the polytope via a maximally safe initial state
Analyzing direction : 1/40
Analyzing direction : 2/40
Analyzing direction : 3/40
Analyzing direction : 4/40
Analyzing direction : 5/40
Analyzing direction : 6/40
```

```
Analyzing direction : 7/40
Analyzing direction : 8/40
Analyzing direction : 9/40
Analyzing direction :10/40
Analyzing direction :11/40
Analyzing direction :12/40
Analyzing direction :13/40
Analyzing direction :14/40
Analyzing direction :15/40
Analyzing direction :16/40
Analyzing direction :17/40
Analyzing direction :18/40
Analyzing direction :19/40
Analyzing direction :20/40
Analyzing direction :21/40
Analyzing direction :22/40
Analyzing direction :23/40
Analyzing direction :24/40
Analyzing direction :25/40
Analyzing direction :26/40
Analyzing direction :27/40
Analyzing direction :28/40
Analyzing direction :29/40
Analyzing direction :30/40
Analyzing direction :31/40
Analyzing direction :32/40
Analyzing direction :33/40
Analyzing direction :34/40
Analyzing direction :35/40
Analyzing direction :36/40
Analyzing direction :37/40
Analyzing direction :38/40
Analyzing direction :39/40
Analyzing direction :40/40
Computing the polytope via the Chebyshev center
Analyzing direction : 1/40
Analyzing direction : 2/40
Analyzing direction : 3/40
Analyzing direction : 4/40
Analyzing direction : 5/40
Analyzing direction : 6/40
Analyzing direction : 7/40
Analyzing direction : 8/40
Analyzing direction : 9/40
Analyzing direction :10/40
Analyzing direction :11/40
Analyzing direction :12/40
Analyzing direction :13/40
Analyzing direction :14/40
Analyzing direction :15/40
Analyzing direction :16/40
Analyzing direction :17/40
Analyzing direction :18/40
Analyzing direction :19/40
```

```
Analyzing direction :20/40
Analyzing direction :21/40
Analyzing direction :22/40
Analyzing direction :23/40
Analyzing direction :24/40
Analyzing direction :25/40
Analyzing direction :26/40
Analyzing direction :27/40
Analyzing direction :28/40
Analyzing direction :29/40
Analyzing direction :30/40
Analyzing direction :31/40
Analyzing direction :32/40
Analyzing direction :33/40
Analyzing direction :34/40
Analyzing direction :35/40
Analyzing direction :36/40
Analyzing direction :37/40
Analyzing direction :38/40
Analyzing direction :39/40
Analyzing direction :40/40
```

Fourier transform (Genz's algorithm and MATLAB's patternsearch)

```
if ft_run
    options = SReachSetOptions('term', 'genzps-open', ...
        'set_of_dir_vecs', set_of_dir_vecs_ft, ...
        'init_safe_set_affine', init_safe_set_affine, 'verbose', 1);
    timer_ft = tic;
    polytope_ft = SReachSet('term','genzps-open', sys,
 prob_thresh, ...
        target_tube, options);
    elapsed_time_ft = toc(timer_ft);
end
```

```
Polytopic underapproximation exists for alpha = 0.80 since W(x_max) =
 0.866.

Analyzing direction : 1/10 | Upper bound of theta: 0.64
OptRAProb |  OptTheta  |  LB_theta  |  UB_theta  |  OptInp^2  | Exit
 reason
  0.8660  |  0.00e+00  |  0.00e+00  |  6.45e-01  |  5.20e-03  |
 Infeasible (0.001)
  0.8050  |  1.61e-01  |  0.00e+00  |  3.22e-01  |  5.78e-03  |
 Feasible
  0.8050  |  2.42e-01  |  1.61e-01  |  3.22e-01  |  8.38e-03  |
 Feasible
  0.8050  |  2.82e-01  |  2.42e-01  |  3.22e-01  |  8.38e-03  |
 Feasible
  0.8050  |  3.02e-01  |  2.82e-01  |  3.22e-01  |  9.20e-03  |
 Feasible
```

```
  0.8050  |  3.12e-01  |  3.02e-01  |  3.22e-01  |  9.20e-03  |
 Feasible
  0.8050  |  3.17e-01  |  3.12e-01  |  3.22e-01  |  9.20e-03  |
 Feasible
Analyzing direction : 2/10 | Upper bound of theta: 0.46
OptRAProb |  OptTheta  |  LB_theta  |  UB_theta  |  OptInp^2  | Exit
 reason
  0.8050  |  2.29e-01  |  0.00e+00  |  4.58e-01  |  4.53e-03  |
 Feasible
  0.8050  |  3.43e-01  |  2.29e-01  |  4.58e-01  |  1.07e-02  |
 Feasible
  0.8050  |  4.00e-01  |  3.43e-01  |  4.58e-01  |  1.09e-02  |
 Feasible
  0.8050  |  4.29e-01  |  4.00e-01  |  4.58e-01  |  1.17e-02  |
 Feasible
  0.8050  |  4.43e-01  |  4.29e-01  |  4.58e-01  |  1.70e-02  |
 Feasible
  0.8050  |  4.50e-01  |  4.43e-01  |  4.58e-01  |  2.13e-02  |
 Feasible
Analyzing direction : 3/10 | Upper bound of theta: 0.56
OptRAProb |  OptTheta  |  LB_theta  |  UB_theta  |  OptInp^2  | Exit
 reason
  0.8050  |  2.78e-01  |  0.00e+00  |  5.56e-01  |  5.14e-03  |
 Feasible
  0.8050  |  4.17e-01  |  2.78e-01  |  5.56e-01  |  8.06e-03  |
 Feasible
  0.8050  |  4.87e-01  |  4.17e-01  |  5.56e-01  |  1.21e-02  |
 Feasible
  0.8050  |  5.22e-01  |  4.87e-01  |  5.56e-01  |  1.21e-02  |
 Feasible
  0.8050  |  5.39e-01  |  5.22e-01  |  5.56e-01  |  1.44e-02  |
 Feasible
  0.8050  |  5.48e-01  |  5.39e-01  |  5.56e-01  |  1.44e-02  |
 Feasible
Analyzing direction : 4/10 | Upper bound of theta: 0.47
OptRAProb |  OptTheta  |  LB_theta  |  UB_theta  |  OptInp^2  | Exit
 reason
  0.8050  |  2.36e-01  |  0.00e+00  |  4.72e-01  |  9.77e-03  |
 Feasible
  0.8050  |  3.54e-01  |  2.36e-01  |  4.72e-01  |  1.09e-02  |
 Feasible
  0.8050  |  4.13e-01  |  3.54e-01  |  4.72e-01  |  1.63e-02  |
 Feasible
  0.8050  |  4.42e-01  |  4.13e-01  |  4.72e-01  |  1.64e-02  |
 Feasible
  0.8050  |  4.57e-01  |  4.42e-01  |  4.72e-01  |  1.55e-02  |
 Feasible
  0.8050  |  4.65e-01  |  4.57e-01  |  4.72e-01  |  1.47e-02  |
 Feasible
Analyzing direction : 5/10 | Upper bound of theta: 0.50
OptRAProb |  OptTheta  |  LB_theta  |  UB_theta  |  OptInp^2  | Exit
 reason
  0.8660  |  0.00e+00  |  0.00e+00  |  5.03e-01  |  5.20e-03  |
 Infeasible (0.001)
```

```
  0.8050  |  1.26e-01  |  0.00e+00  |  2.51e-01  |  1.12e-02  |
 Feasible
  0.8050  |  1.89e-01  |  1.26e-01  |  2.51e-01  |  1.15e-02  |
 Feasible
  0.8050  |  2.20e-01  |  1.89e-01  |  2.51e-01  |  1.15e-02  |
 Feasible
  0.8050  |  2.36e-01  |  2.20e-01  |  2.51e-01  |  1.15e-02  |
 Feasible
  0.8050  |  2.44e-01  |  2.36e-01  |  2.51e-01  |  1.15e-02  |
 Feasible
Analyzing direction : 6/10 | Upper bound of theta: 1.08
OptRAProb |  OptTheta  |  LB_theta  |  UB_theta  |  OptInp^2  | Exit
 reason
  0.8660  |  0.00e+00  |  0.00e+00  |  1.08e+00  |  5.20e-03  |
 Infeasible (0.001)
  0.8660  |  0.00e+00  |  0.00e+00  |  5.39e-01  |  5.20e-03  |
 Infeasible (0.001)
  0.8050  |  1.35e-01  |  0.00e+00  |  2.70e-01  |  9.71e-03  |
 Feasible
  0.8050  |  2.02e-01  |  1.35e-01  |  2.70e-01  |  2.02e-02  |
 Feasible
  0.8050  |  2.36e-01  |  2.02e-01  |  2.70e-01  |  2.02e-02  |
 Feasible
  0.8050  |  2.53e-01  |  2.36e-01  |  2.70e-01  |  2.18e-02  |
 Feasible
  0.8050  |  2.61e-01  |  2.53e-01  |  2.70e-01  |  2.18e-02  |
 Feasible
Analyzing direction : 7/10 | Upper bound of theta: 1.57
OptRAProb |  OptTheta  |  LB_theta  |  UB_theta  |  OptInp^2  | Exit
 reason
  0.8660  |  0.00e+00  |  0.00e+00  |  1.57e+00  |  5.20e-03  |
 Infeasible (0.001)
  0.8660  |  0.00e+00  |  0.00e+00  |  7.83e-01  |  5.20e-03  |
 Infeasible (0.001)
  0.8050  |  1.96e-01  |  0.00e+00  |  3.91e-01  |  1.92e-02  |
 Feasible
  0.8050  |  2.93e-01  |  1.96e-01  |  3.91e-01  |  2.54e-02  |
 Feasible
  0.8050  |  3.42e-01  |  2.93e-01  |  3.91e-01  |  2.57e-02  |
 Feasible
  0.8050  |  3.67e-01  |  3.42e-01  |  3.91e-01  |  3.09e-02  |
 Feasible
  0.8050  |  3.79e-01  |  3.67e-01  |  3.91e-01  |  3.16e-02  |
 Feasible
  0.8050  |  3.85e-01  |  3.79e-01  |  3.91e-01  |  3.06e-02  |
 Feasible
Analyzing direction : 8/10 | Upper bound of theta: 1.38
OptRAProb |  OptTheta  |  LB_theta  |  UB_theta  |  OptInp^2  | Exit
 reason
  0.8660  |  0.00e+00  |  0.00e+00  |  1.38e+00  |  5.20e-03  |
 Infeasible (0.001)
  0.8660  |  0.00e+00  |  0.00e+00  |  6.88e-01  |  5.20e-03  |
 Infeasible (0.001)
```

```
 0.8050  |  1.72e-01  |  0.00e+00  |  3.44e-01  |  1.19e-02  |
 Feasible
 0.8050  |  2.58e-01  |  1.72e-01  |  3.44e-01  |  2.70e-02  |
 Feasible
 0.8050  |  3.01e-01  |  2.58e-01  |  3.44e-01  |  3.45e-02  |
 Feasible
 0.8050  |  3.23e-01  |  3.01e-01  |  3.44e-01  |  3.27e-02  |
 Feasible
 0.8050  |  3.33e-01  |  3.23e-01  |  3.44e-01  |  3.52e-02  |
 Feasible
 0.8050  |  3.39e-01  |  3.33e-01  |  3.44e-01  |  3.41e-02  |
 Feasible
Analyzing direction : 9/10 | Upper bound of theta: 2.11
OptRAProb |  OptTheta   |  LB_theta   |  UB_theta   |  OptInp^2  | Exit
 reason
 0.8660  |  0.00e+00  |  0.00e+00  |  2.11e+00  |  5.20e-03  |
 Infeasible (0.001)
 0.8660  |  0.00e+00  |  0.00e+00  |  1.05e+00  |  5.20e-03  |
 Infeasible (0.001)
 0.8660  |  0.00e+00  |  0.00e+00  |  5.27e-01  |  5.20e-03  |
 Infeasible (0.001)
 0.8050  |  1.32e-01  |  0.00e+00  |  2.64e-01  |  1.25e-02  |
 Feasible
 0.8050  |  1.98e-01  |  1.32e-01  |  2.64e-01  |  1.97e-02  |
 Feasible
 0.8050  |  2.31e-01  |  1.98e-01  |  2.64e-01  |  2.69e-02  |
 Feasible
 0.8050  |  2.31e-01  |  2.31e-01  |  2.64e-01  |  2.69e-02  |
 Infeasible (0.001)
 0.8050  |  2.31e-01  |  2.31e-01  |  2.47e-01  |  2.69e-02  |
 Infeasible (0.001)
Analyzing direction :10/10 | Upper bound of theta: 0.64
OptRAProb |  OptTheta   |  LB_theta   |  UB_theta   |  OptInp^2  | Exit
 reason
 0.8660  |  0.00e+00  |  0.00e+00  |  6.45e-01  |  5.20e-03  |
 Infeasible (0.001)
 0.8050  |  1.61e-01  |  0.00e+00  |  3.22e-01  |  5.78e-03  |
 Feasible
 0.8050  |  2.42e-01  |  1.61e-01  |  3.22e-01  |  8.38e-03  |
 Feasible
 0.8050  |  2.82e-01  |  2.42e-01  |  3.22e-01  |  8.38e-03  |
 Feasible
 0.8050  |  3.02e-01  |  2.82e-01  |  3.22e-01  |  9.20e-03  |
 Feasible
 0.8050  |  3.12e-01  |  3.02e-01  |  3.22e-01  |  9.20e-03  |
 Feasible
 0.8050  |  3.17e-01  |  3.12e-01  |  3.22e-01  |  9.20e-03  |
 Feasible
```

Preparation for Monte-Carlo simulations of the optimal controllers

Monte-Carlo simulation parameters

```
n_mcarlo_sims = 1e5;
n_sims_to_plot = 5;
```

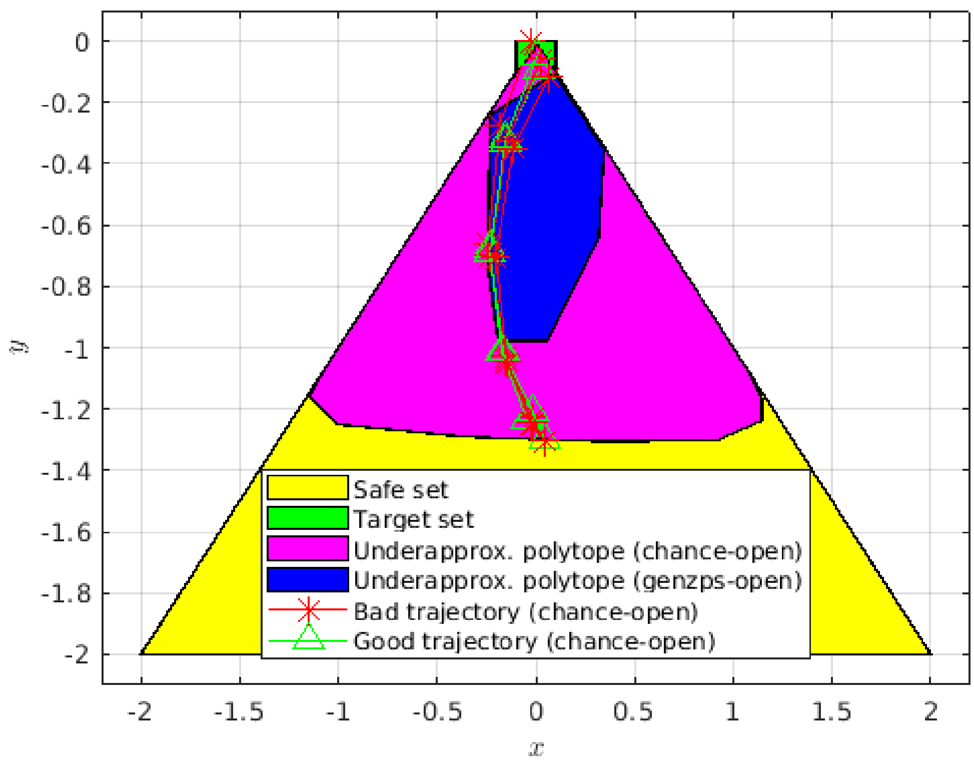
Plotting and Monte-Carlo simulation-based validation

```
figure(1);
clf
box on;
hold on;
plot(safe_set.slice([3,4], slice_at_vx_vy), 'color', 'y');
plot(target_set.slice([3,4], slice_at_vx_vy), 'color', 'g');
legend_cell = {'Safe set','Target set'};
if exist('polytope_cc_open','var')
    plot(polytope_cc_open.slice([3,4],
 slice_at_vx_vy), 'color','m','alpha',1);
    legend_cell{end+1} = 'Underapprox. polytope (chance-open)';
else
    polytope_cc_open = Polyhedron();
    elapsed_time_cc_open = NaN;
end
if exist('polytope_ft','var')
    plot(polytope_ft.slice([3,4],
 slice_at_vx_vy), 'color','b','alpha',1);
    legend_cell{end+1} = 'Underapprox. polytope (genzps-open)';
else
    polytope_ft = Polyhedron();
    elapsed_time_ft = NaN;
end
direction_index_to_plot = 30;
if ~isEmptySet(polytope_cc_open)
    init_state =
 extra_info(2).vertices_underapprox_polytope(:,direction_index_to_plot);
    input_vec =
 extra_info(2).opt_input_vec_at_vertices(:,direction_index_to_plot);
    opt_reach_avoid =
 extra_info(2).opt_reach_prob_i(direction_index_to_plot);

    concat_state_realization = generateMonteCarloSims(...
            n_mcarlo_sims, ...
            sys, ...
            init_state, ...
            time_horizon, ...
            input_vec);

    % Check if the location is within the target_set or not
```

```
    mcarlo_result =
 target_tube.contains([repmat(init_state,1,n_mcarlo_sims);
                                                    concat_state_realization]);
    [legend_cell] = plotMonteCarlo(' (chance-open)',
 mcarlo_result, ...
        concat_state_realization, n_mcarlo_sims, n_sims_to_plot, ...
        sys.state_dim,init_state, legend_cell);
end
legend(legend_cell, 'Location','South');
xlabel('$x$','interpreter','latex');
ylabel('$y$','interpreter','latex');
```



Reporting solution times

```
if any(isnan([elapsed_time_ft, elapsed_time_cc_open]))
    disp('Skipped items would show up as NaN');
end
fprintf('Elapsed time: (genzps-open) %1.3f | (chance-open) %1.3f
 seconds\n', ...
    elapsed_time_ft, elapsed_time_cc_open);
```

Elapsed time: (genzps-open) 1257.204 | (chance-open) 39.743 seconds

Helper functions

Plotting function

```
function [legend_cell] = plotMonteCarlo(method_str, mcarlo_result, ...
    concat_state_realization, n_mcarlo_sims, n_sims_to_plot,
 state_dim, ...
    initial_state, legend_cell)
% Plots a selection of Monte-Carlo simulations on top of the plot

    green_legend_updated = 0;
    red_legend_updated = 0;
    traj_indices = floor(n_mcarlo_sims*rand(1,n_sims_to_plot));
    for realization_index = traj_indices
        % Check if the trajectory satisfies the reach-avoid objective
        if mcarlo_result(realization_index)
            % Assign green triangle as the marker
            markerString = 'g^-';
        else
            % Assign red asterisk as the marker
            markerString = 'r*-';
        end
        % Create [x(t_1) x(t_2)... x(t_N)]
        reshaped_X_vector = reshape(...
            concat_state_realization(:,realization_index), state_dim,
[]);
        % This realization is to be plotted
        h = plot([initial_state(1), reshaped_X_vector(1,:)], ...
                 [initial_state(2), reshaped_X_vector(2,:)], ...
                 markerString, 'MarkerSize',10);
        % Update the legends if the first else, disable
        if strcmp(markerString,'g^-')
            if green_legend_updated
                h.Annotation.LegendInformation.IconDisplayStyle
 = 'off';
            else
                green_legend_updated = 1;
                legend_cell{end+1} = strcat('Good trajectory ',
 method_str);
            end
        elseif strcmp(markerString,'r*-')
            if red_legend_updated
                h.Annotation.LegendInformation.IconDisplayStyle
 = 'off';
            else
                red_legend_updated = 1;
                legend_cell{end+1} = strcat('Bad trajectory ',
 method_str);
            end
        end
    end
end
```

Published with MATLAB® R2017b